REMARKS

Claims 1-33 were pending in the application. Claims 2, 8, 14, and 29-33 have been cancelled. Claims 1, 7, 13, and 24 have been amended. Accordingly, Claims 1, 3-7, 9-13, and 15-28 remain pending in the application.

35 U.S.C. §102 Rejection

Claims 1-33 stand rejected under 35 U.S.C. §102(e) as being anticipated by Beardsley et al. (U.S. Patent Publication No. 2003/0131285).

1.   Applicant respectfully submits that Beardsley fails to teach, "providing a <u>platform editor</u> for making a modification of at least one of said platforms, said test suites, and said execution test harness of said central repository, wherein said modification is <u>automatically applied to all of said clients that are using said at least one of said platforms, said test suites, and said execution test harness</u>" as recited by claim 1.

On pages 3 and 11 of the pending Office Action, the Examiner contends that Beardsley teaches the above-referenced features on page 5, paragraphs 49 and 51 ("determining the test time for the package, if the test time is within a specific range a new package is formed and saved in the database 222"). Applicant respectfully disagrees.

On paragraph 49, Beardsley teaches:

> If there are not pending test packets that can be run on the client computer, then step 802 branches to step 806, where a determination is made <u>whether there are pending test packets that may be run on the client computer 212, 214 with reimaging</u>. That is, although the existing group may not be used, a determination is made whether the computer includes a group that may be used. <u>Using such a group may require rebooting of the client computer 212, 214 to a different operating system and/or installation of software on which to run the tests</u>. (Emphasis added)

On paragraph 51, Beardsley teaches:

> At step 816, the execution time for the package is calculated. If desired, an execution time may be defined at which the test component may not allow the tasks to be performed by the computer. <u>If too long, then step 818 branches back to step 804, where a new package is formed</u>. If not too long, then step 818 branches to step 820, where the changes in the package are committed to the database (e.g., saved in the assigned status file 604). (Emphasis added)

<u>Anticipation</u> requires the presence in a single prior art reference disclosure of <u>each and every element</u> of the claimed invention, <u>arranged as in the claim</u>. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The <u>identical</u> invention must be shown <u>in as complete detail</u> as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Emphasis added).

As noted above, Beardsley teaches determining whether the client computer 212, 214 includes an alternative group that may be used for the pending text packet in the cases where the existing group does not meet the requirements of the pending text packet. If such a group is used it "may require rebooting of the client computer 212, 214 to a different operating system and/or installation of software on which to run the tests" (Beardsley, Page 5, Paragraph 49). Beardsley further teaches calculating "the execution time for the package" and if the execution time is too long "a new package is formed" (Beardsley, Page 5, Paragraph 51). However, Applicant submits that Beardsley fails to teach, "providing a <u>platform editor</u> for <u>making a modification</u> of at least one of said platforms, said test suites, and said execution test harness of said central repository, wherein <u>said modification</u> is **<u>automatically applied to all of said clients that are using said at least one of said platforms, said test suites, and said execution test harness</u>**" as recited by claim 1. In other words, Beardsley fails to teach providing a "platform editor" for modifying "platforms", "test suites", and the "execution test hardness" <u>of the test component 202</u>, "wherein said modification is **<u>automatically applied to all of said clients</u>** using said at least one of said platforms, said test suites, and said execution test harness" (see claim 1).

2.   Furthermore, Applicant submits that Beardsley fails to teach, "responsively to an execution of said installer in said clients, downloading and installing from said central repository selected ones of said platforms and said test suites to said clients for use by said clients in testing said computing products" as recited by claim 1.

On pages 2 and 10-11 of the pending Office Action, the Examiner contends that Beardsley teaches the above-referenced feature of claim 1 on page 3, paragraph 33 and page 5, paragraphs 48 and 53. Applicant respectfully disagrees.

Beardsley teaches:

> Test conditions may be provided to the test component 202 in a number of different ways. In general, the test conditions are provided as tasks that a product developer client 204 would like to be performed in particular platform(s) and language(s). Hereinafter, for ease of discussion, a selected platform and language are referred to herein as a "group." In the embodiment shown in FIG. 2, each product developer client 204 provides a separate test packet 206, 208, 210 for each group on which the product developer wants a product tested. The separate test packet defines tasks that the product developer wants conducted on the product in that group. The number of test packets 206, 208, 210 generated is set by the product developer client 204, and, in the example shown, the product developer client 204_1 provides L test packets, the product developer client 204_2 provides M test packets, and the product developer client 204_3 provides N test packets. A product developer client 204 may provide only one test packet, or may provide several test packets, depending upon the scope of the testing desired. (Beardsley, Page 3, Paragraph 33)

> If a client computer 212, 214 is available, step 700 branches to step 702, where the test component 202 <u>checks to see if the computer is usable</u>. That is, the autolab component 230 determines <u>whether the client computer includes a group and application that meets the requirements of a pending test packet</u>. If not, the process branches back to step 700, where <u>a check is made for other idle client computers 212, 214</u>. If the client computer includes a group and application that meets the requirements of a pending test packet, then step 702 branches to step 704, where the client computer is assigned a test packet (FIG. 5). (Beardsley, Pages 4-5, Paragraph 47) (Emphasis added)

> FIG. 8 shows a general overview of a process for configuring a test packet into a personalized test package for the available client computer 212, 214, and assigning the test package to the client computer, in accordance with one aspect of the present invention. Beginning at step 800, the autolab component 230 checks the state (e.g., what applications are loaded and what group and/or applications

are presently imaged) of the client computer 212, 214. At step 802, a determination is made whether there are pending test packets that can be run on the client computer without the client computer reimaging. That is, whether the tasks of the test packet may be performed on the client computer 212, 214 using the existing group and applications that are imaged by the computer. If so, step 802 branches to step 804, where a personalized test package is built for the client, which may include, for example, preprocessing information and application commands. (Beardsley, Page 5, Paragraph 48) (Emphasis added)

FIG. 9 shows a general overview of the operation of the management component 232 in accordance with one aspect of the present invention. Beginning at step 900, all test packets that have been assigned to a client computer 212, 214 are retrieved that have not already been forwarded to the client computer 212, 214. At step 902, the test packets are broken into subtasks by the management component 232, such as installing an application, performing particular tasks on an application, or the like. The subtasks then are reordered in an efficient manner (step 904). The tasks may be reordered, for example, so that all tasks that are to be performed in a particular application are performed after loading or installing of the application. This process may be performed, for example, by checking the subtasks against a dependency table 608 (FIG. 6) so that they may be efficiently ordered. Additional tasks may be entered (step 906) to fulfill requirements of a computing environment, if needed. The additional tasks may also be maintained in the dependency table 608. (Beardsley, Page 5, Paragraph 53) (Emphasis added)

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Emphasis added)

As noted above, Beardsley teaches determining whether "the computer is usable", that is, "whether the client computer includes a group and application that meets the requirements of a pending test packet." If not, "a check is made for other idle client computers 212, 214." Furthermore, in Paragraph 50, Beardsley teaches, "the client computer 212, 214 is not assigned a test packet" if the client computer "does not include a group and application that may be used."

Beardsley, however, fails to teach, "responsively to an execution of said installer in said clients, downloading and installing **from said central repository** selected ones of said **platforms** and said test suites to said clients for use by said clients in testing said computing products" as recited by claim 1. As noted in Applicant's Specification, "It is an advantage of the system that all necessary information is obtained from a single central location, without need for distributing local copies of the test harness or test suites, which could lead to loss of synchronization." (see Applicant's Specification, page 4, lines 9-13). Beardsley fails to teach downloading and installing from the test component 202 "selected ones of said platforms...for use by said clients in testing" (see claim 1). In fact, as noted above, Beardsley teaches that a client computer 212, 214 is not assigned a test packet if the client computer 212, 214 does not include a group ('group' is defined to mean a selected platform and language, see column 33 of Beardsley) and application that meets the requirements of pending test packets. Furthermore, Beardsley fails to teach performing the "downloading and installing" operation "**responsively** to an execution of said **installer** in said clients" as recited by claim 1.

Additionally, on paragraph 53, Beardsley teaches "test packets are broken into subtasks" and the "subtasks then are reordered in an efficient manner." Specifically, the "tasks may be reordered, for example, so that all tasks that are to be performed in a particular application are performed after loading or installing of the application." While Beardsley teaches that certain tasks or subtasks "that are to be performed in a particular application are performed after loading or installing of the application", Beardsley fails to specifically teach, "**responsively** to an execution of said installer in said clients, **downloading and installing from said central repository** selected ones of **said platforms** and **said test suites** to said clients for use by said clients in testing said computing products" as recited by claim 1.

In accordance, claim 1 is believed to patentably distinguish over Beardsley. Claims 3-6 are dependent upon claim 1 and are therefore believed to patentably distinguish over the cited reference for at least the same reasons.

Likewise, independent claims 7, 13, 19, and 24 recite features similar to those highlighted above with regard to independent claim 1, and are therefore believed to patentably distinguish over the cited reference for at least the reasons given in the above paragraphs discussing claim 1. Claims 9-12 are dependent upon claim 7, claims 15-18 are dependent upon claim 13, claims 20-23 are dependent upon claim 19, and claims 25-28 are dependent upon claim 24. Therefore, claims 8-12, 14-18, 20-23, and 25-28 are believed to patentably distinguish over the cited reference for at least the same reasons.

3.   Applicant also asserts that numerous ones of the dependent claims recite further distinctions over the cited reference. For instance:

4.   Applicant submits that Beardsley fails to teach, "wherein said execution test harness is executed by said clients using binary files thereof residing on said central repository" as recited by claim 3.

On page 11 of the pending Office Action, the Examiner contends that Beardsley teaches the above-referenced feature on page 3, paragraphs 31 and 36. Applicant respectfully disagrees.

Beardsley teaches:

FIG. 2 shows a block diagram of an architecture of a test system 200 incorporating the present invention. In general, the test system 200 of the present invention includes a test component 202 to which <u>a software product developer,</u> such as one of several product developer clients 204_1, 204_2 . . . 204_K, <u>may submit requests for tests on its product, in the form of test packets 206, 208, or 210</u>. The test component 202, in turn, <u>distributes the test packets 206, 208, or 210 to multiple client computers</u> 212_1, 212_2 . . . 212_P, 214_1, 214_2 . . . 214_Q, which <u>conduct the test instructions therein, and return the results to the test component 202</u>. (Beardsley, Page 3, Paragraph 31) (Emphasis added)

The test component 202 includes an application programming interface (API) 220 for <u>receiving the test packets 206, 208, 210</u>. A database 222 is located in, or otherwise is associated with, the test component 202. The database 222 is preferably a relational database, and may be distributed over several machines. As an example, the database may utilize Microsoft Corporation's SQL Server technology, but other database products may be used. (Beardsley, Page 3,

Paragraph 36) (Emphasis added)

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. M.P.E.P 2131; *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). The identical invention must be shown in as complete detail as is contained in the claims. *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Emphasis added)

While Beardsley discloses a software product developer submitting requests for tests on its product, in the form of test packets, to the test component 202, and the test component 202 in turn distributing the test packets to client computers to conduct the tests and return the results to the test component 202, Beardsley fails to teach, "wherein said execution test harness is executed by said clients using binary files thereof **residing on said central repository**" as recited by claim 3. In other words, Beardsley fails to teach "binary files", which are used by the clients to execute the "execution test harness", are located in a central location, e.g., the "central repository", rather than being located at the client computers. As noted in Applicant's Specification, "The binaries themselves are located in only one place, the central repository 12. Centrally locating the binary files is highly advantageous, as only one instance of each binary file need be updated. Furthermore, each user is guaranteed to see the most current version of the framework. Because distribution of local copies of the binaries is avoided, users need not be concerned about having outdated software." (see Applicant's Specification, page 7, lines 22-29; and page 9, lines 5-6).

In accordance, claim 3 is believed to patentably distinguish over Beardsley.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5681-62401.

Respectfully submitted,
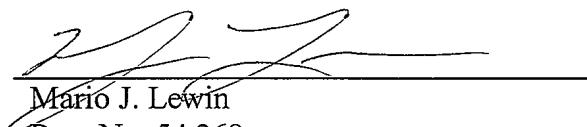
Mario J. Lewin
Reg. No. 54,268
ATTORNEY FOR APPLICANT(S)

Meyertons, Hood, Kivlin,
   Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800
Date: <u>November 21, 2007</u>